UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/843,069	04/26/2001	Rodney Carlton Burnett	AUS920010161	8483

7590 08/08/2006

Darcell Walker
8107 Carvel Lane
Houston, TX 77036

EXAMINER	
SANDOVAL, KRISTIN D	

ART UNIT	PAPER NUMBER
2132	

DATE MAILED: 08/08/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/843,069
Filing Date: April 26, 2001
Appellant(s): BURNETT ET AL.

Darcell Walker, Reg. No. 34,945
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed May 22, 2006 appealing from the Office action mailed October 18, 2005.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,479,612	KENTON ET AL.	12-1995

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. Claims 1-21 rejected under 35 U.S.C. 102(b) as being anticipated by Kenton et al. (Kenton), U.S Patent No. 5,479,612.

As per claim 1, Kenton discloses a method for controlling access to a computer system device comprising steps of:

retrieving the file attributes for the device file used in the system device access attempt (column 3, lines 63-65; column 4, lines 16-24, column 4, lines 41-44);

Kenton demonstrates retrieving file attributes for the device files by obtaining identification information about the device file. In addition, because it has been established that the identification information is being obtained from the device file being used in the system device access attempt, it has also been established that the resource making the access attempt is a device file thus encompassing the second element of this claim.

determining whether the resource that is making the access attempt is a special device file (column 3, lines 63-65; column 4, lines 16-24, column 4, lines 41-44);

As established above, the resource making the access attempt must be a special device file since the claim states that the file attributes will only be retrieved for a device file used on the system device access attempt.

Kenton demonstrates the functionality of a special device file, through a device driver. Device drivers, "act as the portal to the device and its underlying functionality (Background of Invention, paragraph 1, lines 17-18)." Thus, a device driver is a special device file and will be referred to as such for the remainder of this office action.

searching a mapping database for device files that represent the system device that is the object of the access attempt and generating a device file entry list of all protected device files that represent said system device (column 4, lines 29-33; column 5, 18-22);

Kenton exhibits the functionality of a "mapping database" through the use of device identification information as the look up data to be compared to a list of devices supported by the operating system. The identification information is mapped to the device it represents.

Kenton demonstrates the functionality of "protected device files" through the use of device files needing license keys in order to be accessed. Since access is denied if these licenses are not present, this protects the devices from being accessed by the user and are considered protected device files.

generating an authorization decision for the access attempt to the system device based on the security policy that governs each device file in the device file entry list (column 5, lines 36-47).

Unless appellant defines a more specific security policy, the one demonstrated by Kenton, based on the presence of driver licenses, qualifies as a security policy that generates an authorization decision for an access attempt.

As per claim 2, the rejection of claim 1 is incorporated, and further Kenton discloses before said searching step the step of terminating said access control method when the accessing resource is not a special device file (column 4, lines 34-40).

As previously stated in claim 1, the resource must be a device file making the access attempt to have the file attributes retrieved from it, thus, if it were not a device file the file attributes would not have been retrieved and the identification information needed in order to

proceed to the next step of the access control method would not have been obtained. As a result the method would be terminated.

As per claim 3, the rejection of claim 1 is incorporated, and further Kenton discloses after said searching step the step of terminating said access control method when said searching step did not find any database entries that had device specifications that match the device specifications of the device file making the access attempt (column 4, lines 30-40).

Kenton's identification information embodies the functionality of appellant's device specification.

As per claim 4, the rejection of claim 1 is incorporated, and further Kenton disclose said searching step comprising the steps of:

retrieving an entry from the mapping database (column 4, lines 29-34);

comparing the device specification of the device file making the access attempt to the device specification of the database entry (column 4, lines 29-34); and

comparing the file name of the device file making the access attempt to the protected object name of the database entry (column 4, lines 29-34).

Kenton demonstrates the functionality of retrieving an entry from the list, i.e. mapping database, by virtue of the comparison step. In order to find and compare the correct peripheral device in the list, an entry in the list has already been retrieved in order to make the comparison since the entire list cannot be compared at the same time. Kenton shows the comparison of the immediate entry against each entry in the list. Each entry contains the device identification information and the device the identification information represents, thereby showing how this step compares both the specification of the device file and the object name.

As per claim 5, the rejection of claim 4 is incorporated, and Kenton discloses a method further comprising after said file name comparison step the steps of:

generating a device file entry list containing the database entry with the same file specification and file name as the device file making the access attempt (column 5, lines 27-28);

Kenton demonstrates the functionality of generating a device file entry list by writing to a log file.

terminating said searching step (column 5, lines 46-47).

As per claim 6, the rejection of claim 4 is incorporated, and Kenton discloses a method further comprising after said file name comparison step the steps of placing in a file entry list, a mapping database entry having the same file specification as, but different file name from the device file making the access attempt (column 5, lines 36-40).

Kenton shows the functionality of the list the appellant mentions through a list of devices which all share similar attributes and are grouped together but lack a driver license which is another way in which the peripheral devices are identified and access is controlled, i.e. the device file name.

As per claim 7, the rejection of claim 6 is incorporated, and further Kenton discloses a method comprising the steps of:

determining whether there are more entries in the database (column 4, lines 33-36);

retrieving the next mapping database entry for comparison with said device file making the access attempt, when more entries are found in the mapping database (column 4, lines 33-36); and

returning to said device file comparison step (column 4, lines 33-36).

In order to be assured that the a peripheral device is not included in the list, the search must include looping through the entire list entry by entry until no more entries remain.

As per claim 8, the rejection of claim 2 is incorporated, and further Kenton discloses a method wherein said authorization decision step comprises the steps of:

retrieving the current entry in the device file entry list (column 5, lines 18-22);

In order to do the search, an fentry would have to be retrieved in order to proceed to the access decision step.

calling the access decision component to obtain an access decision for the access attempt to the system device based on the security policy that governs the current entry in the device file entry list (figure 2, item 216);

determining whether decision component granted access (column 5, lines 46-47);

The purpose of the access decision component is to decide whether or not to grant the resource access to the device, therefore this step is redundant since it is already incorporated into the access decision component.

determining whether more entries are in this file entry list, if decision component granted access (column 5, lines 36-46); and

updating current entry in said device file entry list and returning to said current entry retrieving step (column 5, lines 36-46).

Kenton exhibits the functionality of looping from the step of retrieving the next entry in the file entry list and determining if there are more entries by having to add all of the values in the quantity fields for every valid installed key. In order to exhaust every valid installed key in the list, this step would have to loop through the entire list to add up each value, therefore, it

would have to determine whether there are more entries and then return to the retrieval step if there were remaining items in the list.

As per claim 9, the rejection of claim 8 is incorporated, and further Kenton discloses comprising after said decision determination step the step of denying the access attempt to the system device if the decision component of a device file entry denies access (item 216, figure 2, follow the "optional no" path).

As per claim 10, the rejection of claim 8 is incorporated, and further Kenton discloses a method comprising the step of allowing the access attempt to the system device if no more entries are in the file entry list (step 216, figure 2).

As previously stated, step 216 exhausts the entire list of valid installed keys in order to find the sum of all entries. Once the sum is computed, there are no more entries in the list and regardless of the decision, both paths lead to the use of the device.

As per claim 11, Kenton discloses a method for controlling access to a computing system device being accessed through a device file, said access control being through an externally stored resource and comprising the steps of:

monitoring the computing system for activities related to creating and accessing special device files that represent system devices (column 3, lines 25-30);

Since device drivers are the communication line between the peripheral devices themselves and the operating system, the device drivers themselves monitor when an access attempt is being made.

restricting the creation of special device files based on rules defined in the externally stored resource (column 4, lines 64-67); and

restricting special device file accesses based on rules defined in the externally stored resource (column 5, lines 5-8).

The special device file access is restricted based on the rules associated with the driver license.

As per claims 12-19, this is a product version of the claimed method discussed above in claims 1-11 wherein all claimed limitations have also been addressed and/or cited as set forth above.

As per claim 20, Kenton discloses a computer connectable to a distributed computing system, which includes special device files containing information, related to corresponding system devices comprising:

a processor (column 3, line 5; item 112, figure 1);

a native operating system (column 3, lines 21-22; item 106, figure 1);

application programs (column 3, lines 57-59);

an externally stored authorization program overlaying said native operating system and augmenting the standard security controls of said native operating system (column 4, lines 41-44);

a mapping database within said external authorization program containing a system device to a protected object name entries for each protected file system object (column 4, lines 29-33);

and

a decision component within said authorization program for controlling access to special device files representing system devices (column 5, lines 15-22; column 6, lines 52-53).

As per claim 21, the rejection of claim 20 is incorporated, and Kenton discloses a computer comprising an authorization program for restricting the creation of special device files representing protected system devices (column 4, lines 64-67).

(10) Response to Argument

The Appellant has argued:

First, Kenton does not incorporate the use of device files in its implementation. As stated in Appellants' disclosure, (paragraph [0002], the file is the fundamental object in a computing system for representing system resources. This representation holds true for attached hardware devices and virtual "pseudo" devices that are represented and accessed through a specialized file type known as a "special device file". The device file acts at the portal to the device and its underlying functionality. This type of file contains no data, but has as part of its attributes, information describing the device.

In response to appellant's argument that the references fail to show certain features of appellant's invention, it is noted that the features upon which appellant relies (i.e., the device file contains no data, but has as part of its attributes, information describing the device.) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). The Examiner contends that Kenton utilizes software device drivers which, as Appellant states, represents system resources such as attached hardware and acts as a portal to the device and its underlying functionality as described in

Kenton, column 3, lines 25-30, "**The operating system 106 includes an internal software driver(s) 122 which permits the host system 102 to: (1) identify external peripheral devices 110; (2) issue commands to peripheral devices 110; (3) process exceptions returned by those peripheral devices 110; and (4) command the transfer of data to and from the peripheral devices 110.**" Therefore, Kenton does incorporate device files in its implementation.

In addition, Examiner contends that Appellant utilizes "device files" and "special device files" interchangeably in the arguments. For example, in the above argument Appellant states, "This representation holds true for attached hardware devices and virtual "pseudo" devices that are represented and accessed through a specialized file type known as a 'special device file'. The **device file** acts at the portal to the device and its underlying functionality. This type of file contains no data, but has as part of its attributes, information describing the device (emphasis added)." It is not clear whether the device file acts as the portal or a special device file acts as a portal.

The Appellant further argues:

The examiner asserts that Kenton exhibits the functionality of a "mapping database" through the use of device identification information as the look up data base to be compared to a list of devices supported by the operating system.

Appellant asserts that Kenton does not describe this searching step.

The Examiner contends that the list searched in Kenton serves as a mapping database since it maps the device identification information to the actual devices supported by the

operating system as stated in Kenton, column 4, lines 30-37, **"In decisional step 204, the operating system 106 examines the device identification information returned from step 202 and compares this data to a list (not shown, but located within the operating system 106) of peripheral devices supported by the operating system 106. If peripheral device 110 is not included in the aforementioned internal list, then the peripheral device 110 is not compatible with the operating system 106."** In order to compare the data to the list, the list would be searched, item by item, iteratively, in order to ensure that the device did or did not exist on the operating system.

The Appellant further argues:

Kenton does not describe generating a special device file entry list of all protected device files that represent a system device.

As stated in column 5, lines 5-9, **"...when a customer desires access of a peripheral device 110 of the type that requires a license, the customer must purchase a software driver license for that peripheral device 110. The corresponding software driver license key 302 is then installed in the keys files 108."** Therefore, a new license is generated and thus a complete entry list of all of the devices protected by a license is generated since the license is for the device driver that protects the device.

The Appellant further argues:

Column 4, lines 29-33 of Kenton describe the comparison of device identification information to a list of peripheral devices supported by the operating system. However, Kenton goes on in that same column 4 to describe the list as an internal list. This list is a predetermined static list that is used for each access attempt. This list is not dynamically generated each time there is an access attempt as is the case in the present invention. Referring to Figure 2 step 23, the access decision step uses the entries in the created list to determine whether or not to grant the access request. This step involves a sort comparison of the list entry with security rules. These security rules are more analogous to the internal list described in Kenton. However, this list is in step 23 and not step 19. As mentioned, Kenton does not generate a dynamic list as described in the mapping step (step 19, Figure 2) of the present invention.

In response to appellant's argument that the references fail to show certain features of appellant's invention, it is noted that the features upon which appellant relies (i.e., the list being dynamically generated each time there is an access attempt in the mapping step) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

The Appellant further argues:

Contrary to the examiner's assertion that all of the elements of claim are discloses in the cited reference (Kenton 5,479,612), the element of 'searching a mapping database for special device files that represent the system device that is the object of the access attempt and

generating a special device file entry list of all protected device files that represent said system device' is not, so the rejection of claim 1 is unsupported by the art and should be withdrawn.

The Examiner contends that the list searched in Kenton serves as a mapping database since it maps the device identification information to the actual devices supported by the operating system as stated in Kenton, column 4, lines 30-37, **"In decisional step 204, the operating system 106 examines the device identification information returned from step 202 and compares this data to a list (not shown, but located within the operating system 106) of peripheral devices supported by the operating system 106. If peripheral device 110 is not included in the aforementioned internal list, then the peripheral device 110 is not compatible with the operating system 106."** In order to compare the data to the list, the list would be searched, item by item, iteratively, in order to ensure that the device did or did not exist on the operating system.

As stated in column 5, lines 5-9, **"...when a customer desires access of a peripheral device 110 of the type that requires a license, the customer must purchase a software driver license for that peripheral device 110. The corresponding software driver license key 302 is then installed in the keys files 108."** Therefore, a new license is generated and thus a complete entry list of all of the devices protected by a license is generated since the license is for the device driver that protects the device.

The Appellant further argues:

Lines 5 through discuss the requiring of a license in order to access a desired peripheral device. This particular step underscores the distinction between Kenton and the present invention. Because the objectives of the inventions are so different, steps that appear to the same or similar are not similar. The present invention contains many more activities than Kenton for Kenton to anticipate the steps in the present invention.

The Examiner contends that it is unclear as to the rationale for Appellant's arguments since Appellant did not identify the, "distinction between Kenton and the present invention." Nor does Appellant specify how, "the objective of the inventions are so different" or how certain "steps that appear to be the same or similar are not similar". Finally, Appellant does not specifically point out the "many more activities" the present invention contains that Kenton does not and it is unclear whether these activities or differences are claimed. Therefore, as best understood, the Examiner assumes that Appellant argues that Kenton does not teach "restricting special device file access based on rules defined in the externally stored resource".

The Examiner contends that external is a relative term and the claims do not specify what the stored resource is external to. Therefore, Examiner interpreted "externally stored resource" as being external to the operating system. Special device file accesses are restricted based on rules contained in the licenses which are externally stored within the Keys file which is item 108 in Figure 1 and is external to the Operating System, item 106.

The Appellant further argues:

First, the database in the present invention is a database contained in an external authorization program. The list in Kenton is an internal list. There is a structural difference between the Kenton list and the mapping database. The fact that the mapping database is externally stored is a key distinction between Kenton and the present invention. The present invention is focus on control using an external and not internal authorization policy. The operations of Kenton appear to be contained within the operating system. In the present invention, the externally stored authorization program overlaying said native operating system. This structural different necessitates the functional/operational differences between Kenton and the present invention.

Again, the Examiner contends that external is relative and Appellant did not specify what the authorization program was external to. As stated in column 5, lines 5-9, **"...when a customer desires access of a peripheral device 110 of the type that requires a license, the customer must purchase a software driver license for that peripheral device 110. The corresponding software driver license key 302 is then installed in the keys files 108."** Therefore, the licenses, which authorize a user to access the device files in order to access the device, are stored and downloaded through an external program from the operating system. The licenses must be stored where this authorization program is located in order to be downloaded. In addition, the keys file in Kenton is maintained and stored externally to the operating system as seen in Figure 1. In addition, the keys file is a mapping database since it maps driver license keys to their corresponding class of peripheral devices as seen in column 3, lines 51-55. The use

of license overlays the operating system since the operating system checks for the licenses as seen in column 5, lines 18-22.

Finally, Appellant argues:

One situation that occurs in the present invention that is not in Kenton is when more entries are found during a search. Kenton is basically a one-to-one matching. There is no process where iterative process which requires an acceptance of all entries on the generated list before access is granted. Because the dynamically generated list can have multiple entries, there are steps in the dependent claims that address the iterative process of evaluating each entry on a generated list. There are also steps in the dependent claims that cover the terminating of these iterative processes.

In response to appellant's argument that the references fail to show certain features of appellant's invention, it is noted that the features upon which appellant relies (i.e., requiring an acceptance of all entries on the generated list before access is granted and the list being dynamically generated each time there is an access attempt in the mapping step) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

As stated in column 3, lines 51-55, **"The operating system 106 performs this task by searching through the keys file 108 for any and all driver license keys 302 that correspond to the class of peripheral devices which includes the peripheral devices 110 in question."**

Therefore, since the operating system searches "for any and all driver license keys" it must go through the keys file list iteratively, in addition, it is not a one-to-one matching since it states "any and all" which means there could be more then one driver license key corresponding to a class of peripheral devices.

Appellant states that, "Because of the clear distinctions between in the independent claims between Appellants' invention and the cited reference Kenton, Appellants do not discuss in detail the distinctions in the dependent claims (pg. 9, 2nd paragraph of Appellant's brief)." Therefore it is ambiguous as to which dependent claims Appellant is referring to when stating, "There are also steps in the dependent claims that cover the terminating of these iterative processes." However, the Examiner points to column 5, lines 23-29, **"If no valid, matching license key 302 is found, then in the preferred embodiment, the operating system 106 displays a licensing violation message instructing the user to obtain a driver license key 302 in order to access the peripheral device 110 which requires a license, shown in step 214. Additionally, in step 214, the operating system 106 stores a record of the violation in the log file 118."** Therefore, the searching terminates when a valid driver license key is not found and it has already been established that the search is iterative.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

(12) Conclusion

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

KDS
KDS 8/1/2006

Conferees:

Kim Vu KV

Chris Revak

8/3/06
CHRISTOPHER REVAK
PRIMARY EXAMINER